



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/015,958	10/30/2001	David George De Vorchik	MFCP,88142	6989
45809                      7590                      06/23/2008 SHOOK, HARDY & BACON L.L.P. (c/o MICROSOFT CORPORATION) INTELLECTUAL PROPERTY DEPARTMENT 2555 GRAND BOULEVARD KANSAS CITY, MO 64108-2613				
EXAMINER				
KISS, ERIC B				
ART UNIT		PAPER NUMBER		
2192				
MAIL DATE		DELIVERY MODE		
06/23/2008		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents  
United States Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

**BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES**

Application Number: 10/015,958  
Filing Date: October 30, 2001  
Appellant(s): DE VORCHIK ET AL.

\_\_\_\_\_  
Monplaisir Hamilton (Reg. No. 54,851)  
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed August 3, 2007 (as corrected in the supplemental filing on September 28, 2007), appealing from the Office action mailed December 5, 2006.

**(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief (**as replaced by appellant on September 28, 2007**) is correct. Note that appellant's heading in the corrected version should read "III. STATUS OF CLAIMS", but the replacement section otherwise appears correct.

**(4) Status of Amendments After Final**

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

**(8) Evidence Relied Upon**

6,574,791                      GAUTHIER et al.                      6-2003

Alex Fedorov, et al., Professional Active Server Pages 2.0, 1998, Wrox Press Ltd., pp. i, ii, 405-432.

**(9) Grounds of Rejection**

The following ground(s) of rejection are applicable to the appealed claims:

**A)      Claims 7, 10, and 11 are rejected under 35 U.S.C. 102(b) as being anticipated by Alex Fedorov, et al., “Professional Active Server Pages 2.0,” 1998, Wrox Press Ltd (hereinafter *Fedorov et al.*).**

As per claim 7, *Fedorov et al.* discloses generating a host wizard that defines an extension interface to respond to navigation events (*seismic.asp*, an Active Server Page that causes the loading and display of a wizard dialog box; see, for example, the last paragraph on p. 423; the response to navigation events is implemented through the “< Back”, “Next >”, “Cancel”, and “Send” navigation controls as illustrated, for example, on pp. 424-426); generating a web component comprising: a web page containing a header area, a wizard control area and a control interface area (see, for example, the screenshots of the wizard dialog on pp. 424-426), the control interface area having navigation control to recursively navigate within said web component and to said host wizard, by utilizing one or more object module functions enabling navigation (the screenshots of the wizard dialog on pp. 424-426 clearly show the “< Back” and “Next >” navigation controls). *Fedorov et al.* further discloses generating a user interface that integrates the web component into the host wizard by utilizing the extension interface to perform recursive navigation between said web component and said host wizard (the user-interface is

provided by *equakeget.htm*, an HTML page, which interacts with the user and submits entered results to the Active Server Page script in *seismic.asp*; see, for example, “Submitting the Data” on p. 431); and utilizing an information container to exchange informational items between the web component and the host wizard (a return string is generated to convey the results; see, for example, “Submitting the Data” on p. 431).

As per claim 10, this is a computer readable medium version of the method discussed above (claim 7). The use of such a computer readable medium, such as memory, is further inherent in realizing the computer-implemented functionality disclosed by *Fedorov et al.*

As per claim 11, this is a computer system version of the method discussed above (claim 7). *Fedorov et al.* further discloses the prescribed methods as being computer-implemented (for example, the screenshots on pp. 424-427 illustrate execution within an Internet Explorer web browser environment, which inherently requires a processor and a memory to function as illustrated/described).

**B) Claims 1-3, 8, 9, and 14 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,574,791 to Gauthier et al.**

As per claim 1, *Gauthier et al.* discloses generating a host-wizard component (see, for example, col. 9, line 56, through col. 10, line 5); generating one or more sub-wizard components (see, for example, col. 9, line 56, through col. 10, line 5); and said host-wizard invoking said one or more subwizard components during said host-wizard component execution (see, for example, col. 9, line 56, through col. 10, line 5); and transferring control from said host-wizard to said one or more sub-wizard components. As disclosed in col. 14, line 9, through col. 15, line 24, the

subwizard includes several objects that define its functionality. For example, the WizardState object included in the subwizard maintains a set of attributes used to get and channel data from default attributes and/or user input and deliver those attributes to a WizardCodeGenerator object (col. 14, lines 29-37), and the WizardCodeGenerator object uses this channeled data to either generate code (for example, in the context of a wizard designed to configure a peripheral device) or perform or execute existing code (col. 15, lines 1-24). As the functionality for the subwizard is defined within the subwizard, control must be passed to the subwizard during execution in order to realize the described functionality.

As per claim 2, *Gauthier et al.* further discloses the sub-wizard components being browser based object components (see, for example, col. 18, line 66, through col. 19, line 30).

As per claim 3, *Gauthier et al.* further discloses the sub-wizard components being operating system based application component object extensions (the basic functionality of the computer system disclosed by *Gauthier et al.* is controlled by operating system 100; see, for example, col. 6, lines 42-46).

As per claim 8, this is a computer readable medium version of the method discussed above (claim 1). *Gauthier et al.* further discloses the use of such a medium to implement the prescribed methods (see, for example, col. 5, line 52, through col. 6, line 4).

As per claim 9, this is a computer system version of the method discussed above (claim 1). *Gauthier et al.* further discloses the use of such a system to implement the prescribed methods (see, for example, col. 5, line 24, through col. 6, line 46).

As per claim 14, this is a computer system version of the method discussed above (claim 12). *Gauthier et al.* further discloses the use of such a system to implement the prescribed methods (see, for example, col. 5, line 24, through col. 6, line 46). *Gauthier et al.* also discloses the wizards having panels to guide a user through tasks (see, for example, col. 2, lines 4-13; col. 11, lines 54-63).

**C) Claims 16, 18, and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,574,791 to Gauthier et al. in view of Alex Fedorov, et al., “Professional Active Server Pages 2.0,” 1998, Wrox Press Ltd (“*Fedorov et al.*”).**

As per claims 16, 18, and 19, although *Gauthier et al.* discloses the features recited in claims 1, 8, and 9 (see the rejection under 35 U.S.C. § 102(b) above), *Gauthier et al.* fails to expressly disclose passing a property bag (an assorted collection of miscellaneous data, variables and other information that a developer needs to transfer between wizards) between said host-wizard component and said one or more sub-wizard components. However, *Fedorov et al.* teaches providing a host wizard (*seismic.asp*, an Active Server Page that causes the loading and display of a wizard dialog box; see, for example, the last paragraph on p. 423) and a subwizard (the user-interface provided by *equakeget.htm*, an HTML page, which interacts with the user and submits entered results to the Active Server Page script in *seismic.asp*; see, for example, “Submitting the Data” on p. 431), wherein a “property bag” (a return string) is generated to pass the results back to the host wizard (see, for example, “Submitting the Data” on p. 431, describing the passing of collected data to *seismic.asp* and the figure on p. 427, illustrating the subsequent display of the generated output in the web browser window). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the

Art Unit: 2191

wizards/subwizards of *Gauthier et al.* to include such passing of data (a “property bag”) as per the teachings of *Fedorov et al.* One would be motivated to do so to allow a host to process the data gathered by a sub-wizard to accomplish an overall task, particularly where large amounts of data must be collected or when tasks are complex (see, for example, *Fedorov* “Using Enhanced Forms with ASP” and “A Tour of the Seismic Load Calculator” on p. 423).



**(10) Response to Argument**

**A) Rejection under 35 U.S.C. § 102(b) over Fedorov et al. (Professional Active Server Pages 2.0, 1998, Wrox Press Ltd) (Brief at pp. 12-15)**

Claims 7, 10, and 11

Appellant urges that, “the Office’s interpretation of ‘wizard’ as utilized in claims 7, 10, and 11 and defined in the specification, at p. 2, ll. 7-10, goes beyond the reasonable ordinary and customary meaning of the term as understood by an artisan of ordinary skill.” (Brief at p. 12.) However, this allegation is not supported by any evidence of record. Appellant has not submitted any evidence that suggests any supposed ordinary and customary definition of “wizard” that excludes the examiner’s interpretation of the term in the applied rejections. The examiner has repeatedly invited appellant to clarify the claims by copying text from the specification into the claims if appellant desires specific limitations on how “wizard” is to be interpreted. (Advisory Action, Nov. 4, 2005, item 11; Non-Final Rejection, June 26, 2006, p. 2; Final Rejection, Dec. 5, 2006, p. 3). Appellant has declined to amend, relying instead on mere attorney argument that *Fedorov et al.* does not disclose a wizard. Moreover, the examiner submits that the *Fedorov et al.* reference clearly supports the examiner’s interpretation as entirely consistent with the reasonable ordinary and customary meaning of “wizard” because it explicitly uses the term wizard:

Since the number and complexity of the variables is quite large, it makes sense to offer the inputting of these values in a ‘wizard’ fashion. . . . .  
The other advantage to using a ‘wizard’ is that validating the user’s input is simplified. Within each pane of the wizard we can validate the user’s input before proceeding.

See, e.g., Fedorov et al. at p. 423 (emphasis added). Furthermore, because the wizard of *Fedorov et al.*, **directs a user through a configuration process or the implementation of a particular task** (specifically, the task of calculating seismic base shear, the force that an earthquake applies to a building, *Id.*), is **essentially a programmatic method of providing guidance to a user within a controlled environment and in a predictable manner** (*Id.* (the calculation is based on a series of building parameters that the user specifies and the user's input is validated within each pane of the wizard)), and **includes a multistep process that is controlled by a user's navigation of screens to answer questions and ultimately complete an operation** (*Id.* at pp. 424-427 (illustrating screenshots of the wizard in action)), the wizard of *Fedorov et al.* meets the various "definitions" in appellant's brief. (Brief at p. 13, paragraph 2).

Regarding claims 7, 10, and 11, *Fedorov et al.* discloses generating a user interface that integrates the web component into the host wizard by utilizing the extension interface to perform recursive navigation between said web component and said host wizard (the user-interface is provided by *equakeget.htm*, an HTML page, which interacts with the user and submits entered results to the Active Server Page script in *seismic.asp*; see, for example, "Submitting the Data" on p. 431). The host wizard (*seismic.asp*) is an Active Server Page that causes the loading and display of a wizard dialog box (see, for example, the last paragraph on p. 423), and the response to navigation events is implemented through the "< Back", "Next >", "Cancel", and "Send" navigation controls as illustrated, for example, on pp. 424-426. **The interaction between *equakeget.htm* and *seismic.asp*, facilitated through the navigation buttons, provides the integration of the host wizard and web component.**

Appellant argues that, “Seismic.asp is not a multistep process that is controlled by a user’s navigation of screens.” (Brief at p. 14.) The examiner disagrees. The multistep process of *Fedorov et al.* begins with the user connecting to seismic.asp, and during the initial connection, the user may provide input data through the URL. *Fedorov et al.* at p. 423. Next, seismic.asp displays a loading screen, providing the user with instructions regarding what to do if the subsequent dialog window does not display correctly (*i.e.*, close the dialog window and click the Forward button on the browser to reload the page). *Id.* at p. 424 (the first figure on the page, illustrating the initial instruction to the user from *seismic.asp* and the initial display of the *equakeget.htm* dialog). Next, the initial dialog box corresponding to *equakeget.htm* is displayed, and the user navigates through the various panels using the “< Back”, “Next >”, and “Cancel” buttons. *Id.* at pp. 424–426. On the last panel, the “Cancel” button is replaced with a “Send” button, and clicking on the “Send” button causes the *closedialog()* function to place the values of all of the form elements in the wizard into the return string and closes the dialog box, upon which *seismic.asp* reads the return string and redirects the browser to itself so that the data can be loaded into the server. *Id.* at p. 431. **Thus, the “Send” button of *equakeget.htm* provides recursive navigation between the host wizard (*seismic.asp*) and the web component (*equakeget.htm*) by redirecting the browser back to *seismic.asp* (with the return data contained in the URL).** Finally, once the data has been submitted to the server via the URL, *seismic.asp* returns the results for display to the user. *Id.* at pp. 431–432 (Creating the Result).

**B) Rejection under 35 U.S.C. § 102(e) over Gauthier et al. (U.S. Patent No. 6,574,791)  
(Brief at pp. 16-21)**

Claims 1, 8, and 9 (Brief at pp. 16-17)

Regarding claims 1, 8, and 9, *Gauthier et al.* discloses said host-wizard invoking said one or more subwizard components during said host-wizard component execution (see, for example, col. 9, line 56, through col. 10, line 5); and transferring control from said host-wizard to said one or more sub-wizard components. As disclosed in col. 14, line 9, through col. 15, line 24, the subwizard includes several objects that define its functionality. For example, the WizardState object included in the subwizard maintains a set of attributes used to get and channel data from default attributes and/or user input and deliver those attributes to a WizardCodeGenerator object (col. 14, lines 29-37), and the WizardCodeGenerator object uses this channeled data to either generate code (for example, in the context of a wizard designed to configure a peripheral device) or perform or execute existing code (col. 15, lines 1-24). **As the functionality for the subwizard is defined within the subwizard, control must be passed to the subwizard during execution in order to realize the described functionality.** More specifically, *Gauthier et al.* discloses the subwizards as executable components (*e.g.*, *Gauthier et al.* at col. 14, line 6) and each executable subwizard is self-contained with regard to its panel attributes (entry fields, selection lists, and other GUI elements that allow the target wizard user to input and/or select data) and panel flow (the list of panels that comprise the subwizard and the corresponding panel flow order). *Id.* at col. 14, lines 11-28. Further, the executable functionality necessary to get data from the user interface and channel that data is contained within the subwizards themselves (in the WizardState object of each subwizard). *Id.* at col. 14, lines 29-37. In object-oriented

technology, calling a method (such as the get and set methods defined within the subwizard WizardState object) during execution of a program requires devoting processor resources to such method code, thus passing control to the executable code in such a method. If this were not the case, the executable subwizards of *Gauthier et al.* would never be executed, and their functionality never realized as part of an operable system.

Claim 2 (Brief at pp. 17-19)

Regarding claim 2, *Gauthier et al.* further discloses the sub-wizard components being browser based object components. *See, for example, Gauthier et al.*, col. 18, line 66, through col. 19, line 30 (describing the wizard metadata and the use of XML to represent such). The examiner notes that the term "browser based object component" is found within the original specification only in the context of original claims 2 and 5. **No definition of "browser based object component" that excludes XML is found in the specification (including original claims 2 and 5), nor has appellant advanced such an exclusive definition.** In response to appellant's argument originally presented May 16, 2005 (Remarks, May 15, 2005, pp. 12-13), the examiner had cited evidence that XML (as disclosed in *Gauthier et al.*) is browser-based (*i.e.*, associated with or related to browser technology). (Final Rejection, Aug. 19, 2005, p. 4 (citing *Extensible Markup Language (XML) 1.0 (Third Edition): W3C Recommendation 04 February 2004* ("[The goal of XML] is to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML."))). Appellant had responded to this evidence by merely challenging the prior art status of the XML specification rather than addressing the technical merits (Remarks, Oct. 19, 2005, p. 13, paragraph 2). In a subsequent Advisory Action, the examiner cited an earlier publication of the XML specification (published

February 10, 1998) and again asserted that based on the evidence the XML components of *Gauthier et al.* may reasonably be considered "browser-based". (Advisory Action, Nov. 4, 2005, item 11 (noting that the February 1998 version of the XML specification contained the same evidence as the February 2004 version). Appellant has yet to address this cited evidence.

Claim 3 (Brief at pp. 19-20)

Regarding claim 3, as the wizard components described in *Gauthier et al.* are designed to execute on top of an operating system and provide additional functionality (e.g., enabling the display and interaction with user interface panels designed to facilitate code generation or execution), these components can be interpreted as operating system based extensions. *See, for example, Gauthier et al.*, col. 6, lines 42-46 (describing the operating system as part of a computer system in accordance with a preferred embodiment). **No definition of "operating system based" that excludes the examiner's interpretation of claim 3 (and the application of the *Gauthier et al.* reference) is found in the specification (including the original claims and drawings), nor has appellant advanced such an exclusive definition.**

Claim 14 (Brief at pp. 20-21)

Regarding claim 14, as noted above (see the arguments above regarding claim 1), *Gauthier et al.* discloses transferring control from a first wizard to a second wizard (one or more sub-wizards). The subwizards of *Gauthier et al.* make up the target wizard (col. 7, lines 13-19). *Gauthier et al.*, at col. 13, lines 27-58, further describes the **hierarchical control between the target wizard and its subwizards**. *Gauthier* clearly discloses the use of buttons, such as BACK, NEXT, FINISH, CANCEL, and HELP, to navigate between panels in the target wizard. *Gauthier et al.* at col. 13, line 65, through col. 14, line 8. *Gauthier* further discloses the

use of the GUI panel to allow the user to select one of the subwizards for execution, thus navigating between the host wizard and the subwizard (see, e.g., col. 14, lines 1-28, describing the GUI panels used by the subwizards, including the buttons).

**C) Rejection under 35 U.S.C. § 103(a) over Fedorov et al. (Professional Active Server Pages 2.0, 1998, Wrox Press Ltd) and Gauthier et al. (U.S. Patent No. 6,574,791) (Brief at pp. 21-23)**

Claims 16, 18, and 19

Regarding claims 16, 18, and 19, a “property bag” is interpreted as an assorted collection of miscellaneous data, variables and other information that a developer needs to transfer between wizards. *Fedorov et al.* teaches providing a host wizard (*seismic.asp*, an Active Server Page that causes the loading and display of a wizard dialog box; see, for example, the last paragraph on p. 423 of *Fedorov et al.*) and a subwizard (the user-interface provided by *equakeget.htm*, an HTML page, which interacts with the user and submits entered results to the Active Server Page script in *seismic.asp*; see, for example, “Submitting the Data” on p. 431 of *Fedorov et al.*), wherein a “property bag” (a return string) is generated to pass the results back to the host wizard (see, for example, *Fedorov et al.*, “Submitting the Data” on p. 431, describing the passing of collected data to *seismic.asp* and the figure on p. 427 of *Fedorov et al.*, illustrating the subsequent display of the generated output in the web browser window).

Regarding appellant’s argument the *seismic.asp* (of *Fedorov et al.*) is not a host wizard (Brief at p. 23), please see the response above regarding claims 7, 10, and 11, wherein the multistep process of *seismic.asp* is discussed in detail.

Regarding appellant's argument that *Gauthier et al.* maintains centralized execution of the wizards so information does not need to be passed (Brief at p. 23), the examiner disagrees. See the response above regarding claims 1, 8, and 9, wherein separate execution of the subwizards is discussed in detail. Further, as noted above regarding claims 1, 8, and 9, in the system of *Gauthier et al.*, the subwizards each store and channel data associated with the subwizards (in their respective WizardState objects). See, e.g., *Gauthier et al.* at col. 14, lines 29-37.



**(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

Art Unit: 2191

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Eric B. Kiss/

Eric B. Kiss

Primary Examiner, Art Unit 2192

Conferees:

/Tuan Q. Dam/

Tuan Dam

Supervisory Patent Examiner, Art Unit 2192

/Wei Y.Zhen/

Wei Zhen

Supervisory Patent Examiner, Art Unit 2191